Обработка последовательностей и модели внимания

К. В. Воронцов
vokov@forecsys.ru

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Машинное обучение (курс лекций, К.В.Воронцов)»

26 февраля 2021 • МФТИ

Содержание

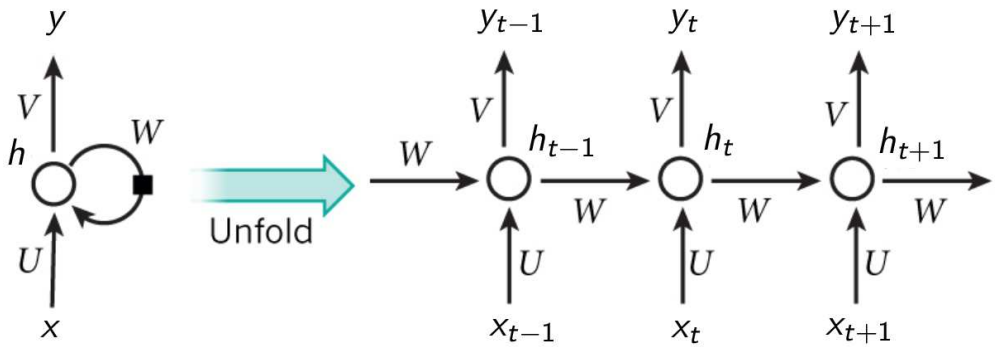
Напоминание. Рекуррентная сеть (RNN)

x_t — входной вектор в момент $t = 1, \ldots, T$

y_t — выходной вектор (в некоторых приложениях $y_t \equiv h_t$)

h_t — вектор скрытого состояния в момент t

$$h_t = \sigma_h(Ux_t + Wh_{t-1})$$
$$y_t = \sigma_y(Vh_t)$$



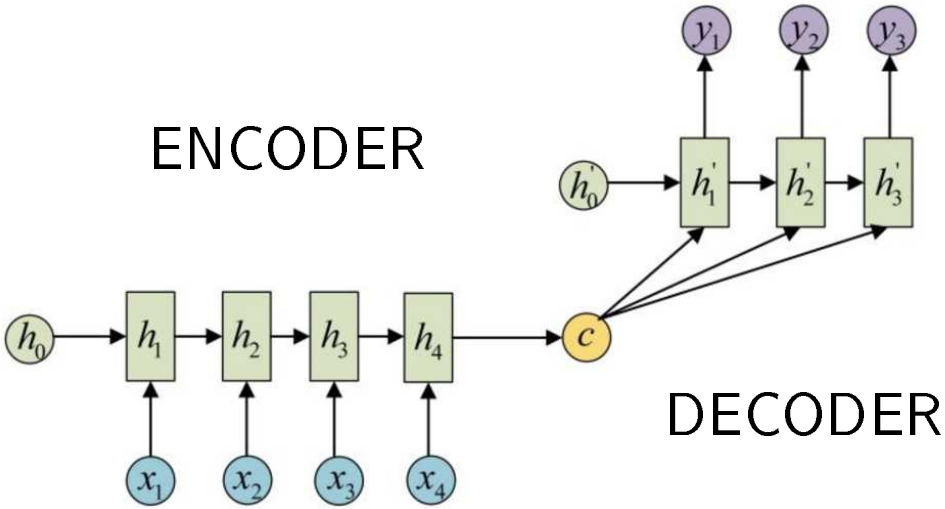
Обучение рекуррентной сети: $\sum_{t=0}^{T} \mathscr{L}_t(U, V, W) \to \min_{U,V,W}$

- длины входного и выходного сигнала обязаны совпадать
- невозможно заглядывание вперёд
- не подходит для многих задач (MT, QA и др.)

Рекуррентная сеть для обработки последовательностей (seq2seq)

$\{x_i : i = 1, \ldots, n\}$ — входная последовательность
$\{y_t : t = 1, \ldots, m\}$ — выходная последовательность
$c \equiv h_n$ кодирует всю информацию про $\{x_i\}$ для синтеза $\{y_t\}$

$$h_i = f_{\text{in}}(x_i, h_{i-1})$$
$$h'_t = f_{\text{out}}(h'_{t-1}, y_{t-1}, c)$$
$$y_t = f_y(h'_t, y_{t-1})$$



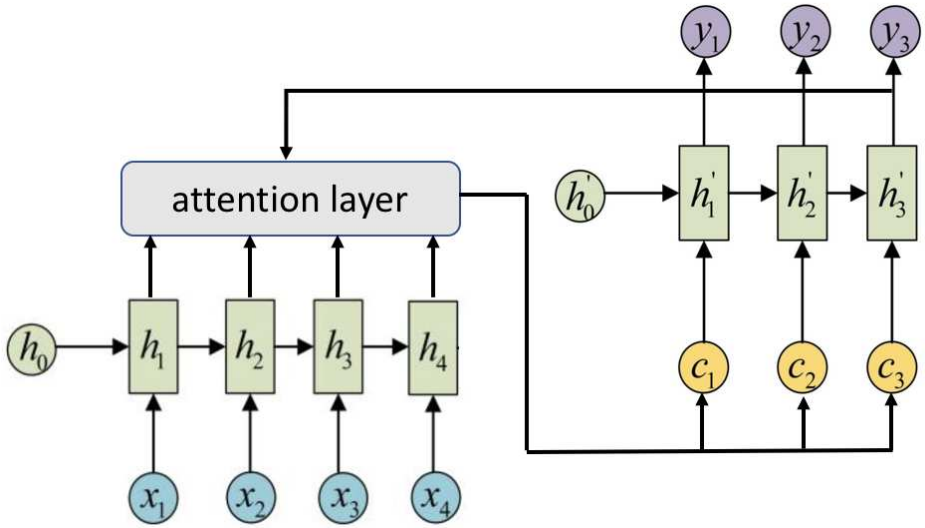
- h_n лучше помнит конец последовательности, чем начало
- чем больше n, тем труднее упаковать всю информацию в c
- придётся контролировать затухание/взрывы градиента
- RNN трудно распараллеливается

Рекуррентная сеть с вниманием (attention mechanism)

$a(h, h')$ — функция сходства состояний входа h и выхода h'
α_{ti} — важность входа i для выхода t (attention score), $\sum_i \alpha_{ti} = 1$
c_t — вектор входного контекста для выхода t (context vector)

$$h_i = f_{\text{in}}(x_i, h_{i-1})$$
$$\alpha_{ti} = \underset{i}{\text{norm}}\, a(h_i, h'_{t-1})$$
$$c_t = \sum_i \alpha_{ti} h_i$$
$$h'_t = f_{\text{out}}(h'_{t-1}, y_{t-1}, c_t)$$
$$y_t = f_y(h'_t, y_{t-1}, c_t)$$

здесь и далее $\underset{i}{\text{norm}}(p_i) = \frac{p_i}{\sum_k p_k}$



- можно отказаться от рекуррентности как по h_i, так и по h'_t
- можно вводить обучаемые параметры в a и c

Bahdanau et al. Neural machine translation by jointly learning to align and translate. 2014.

Применения моделей внимания

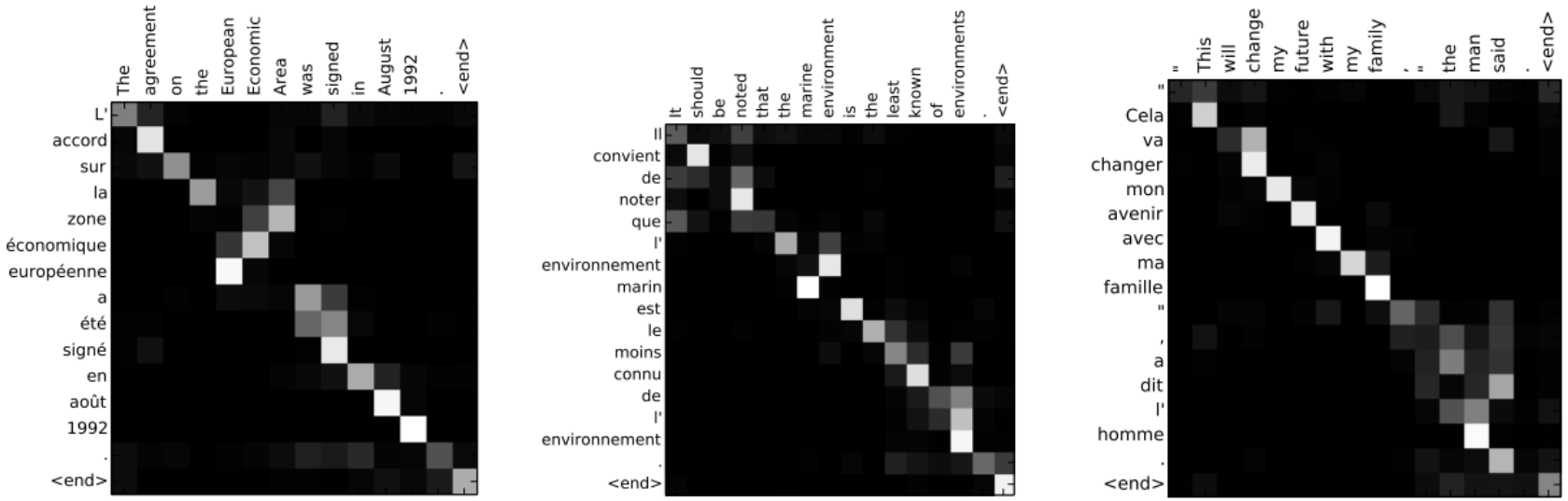
Преобразование одной последовательности в другую, seq2seq:

- Машинный перевод (machine translation)
- Ответы на вопросы (question answering)
- Суммаризация текста (text summarization)
- Описание изображений, аудио, видео (multimedia description)
- Распознавание речи (speech recognition)
- Синтез речи (speech synthesis)

Обработка последовательности:

- Классификация текстовых документов
- Анализ тональности документа / предложений / аспектов

Применения моделей внимания в машинном переводе



Интерпретируемость моделей внимания:

При обработке конкретной последовательности x визуализация матрицы α_{ti} показывает, на какие слова x_i модель обращает внимание, генерируя слово перевода y_t

---

Bahdanau et al. Neural machine translation by jointly learning to align and translate. 2014.

Модели внимания на изображениях для генерации описаний

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

При генерации каждого слова в описании изображения визуализация показывает, на какие области изображения модель обращает внимание, генерируя данное слово

Kelvin Xu et al. Show, attend and tell: neural image caption generation with visual attention. 2016

Разновидности функций сходства векторов

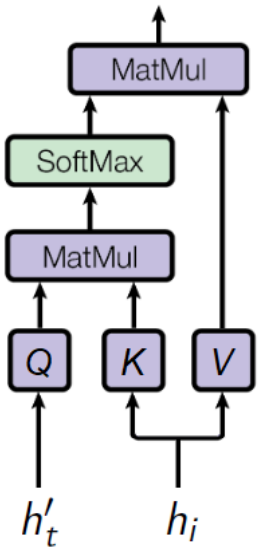
$a(h, h') = h^\mathsf{T} h'$ — скалярное произведение
$a(h, h') = \exp(h^\mathsf{T} h')$ — тогда norm превращается в SoftMax
$a(h, h') = h^\mathsf{T} W h'$ — обобщение, с матрицей параметров W
$a(h, h') = w^\mathsf{T} \text{th}(Uh + Vh')$ — аддитивное внимание с (w, U, V)

Обобщение с тремя матрицами Query, Key, Value:

$$a(h_i, h'_{t-1}) = (Kh_i)^\mathsf{T}(Qh'_{t-1})$$
$$\alpha_{ti} = \text{SoftMax}_i\, a(h_i, h'_{t-1})$$
$$c_t = \sum_i \alpha_{ti} V h_i$$

где $Q_{d\times\dim(h')}$, $K_{d\times\dim(h)}$, $V_{d\times\dim(h)}$ — матрицы весов линейных нейронов (обучаемые линейные преобразования в пространство размерности d)

Возможно упрощение модели: $K \equiv V$



Vaswani et al. Attention is all you need. 2017.
Dichao Hu. An introductory survey on attention mechanisms in NLP problems. 2018.

Многомерное внимание (multi-head attention)

Идея: несколько разных моделей совместно обучаются обращать внимание на разные аспекты входной информации

Вычисляется K функций сходства $a^k(h, h')$, $k = 1, \ldots, K$:

$$a^k(h_i, h'_{t-1}) = h_i^{\mathsf{T}} W^k h'_{t-1}$$
$$\alpha_{ti}^k = \mathrm{SoftMax}_i\, a^k(h_i, h'_{t-1})$$
$$c_t^k = \sum_i \alpha_{ti}^k V^k h_i$$

Два варианта агрегирования выходного вектора:

$c_t = \frac{1}{K} \sum_{k=1}^K c_t^k$ — усреднение

$c_t = \left[c_t^1 \cdots c_t^K\right]$ — конкатенация

$c_t = \left[c_t^1 \cdots c_t^K\right] W_c$ — чтобы сохранить размерность $\dim c_t = d$

Предсказание по агрегированному вектору:

$$y_t = f_y(h'_t, y_{t-1}, c_t)$$

Vaswani et al. Attention is all you need. 2017.

Dichao Hu. An introductory survey on attention mechanisms in NLP problems. 2018.

Иерархическое внимание (hierarchical attention)

Вложенная структура: *слова* $\in$ *предложения* $\in$ *документы*

x_{it} — слова $t = 1, \ldots, T_i$ в предложениях $i = 1, \ldots, L$

Сеть первого (нижнего) уровня, обучение эмбедингов s_i:

$h_{it} = \text{BidirGRU}(W_0 x_{it})$ — GRU для векторизации слов

$u_{it} = \text{th}(W_1 h_{it} + b_1)$ — обучаемое преобразование Key

$s_i = \sum_t h_{it} \text{SoftMax}_t(u_{it}^\text{T} q_1)$ — эмбединг предложения, Query q_1

Сеть второго (верхнего) уровня, обучение эмбедингов v:

$h_i = \text{BidirGRU}(s_i)$ — GRU для векторизации предложений

$u_i = \text{th}(W_2 h_i + b_2)$ — обучаемое преобразование Key

$v = \sum_i h_i \text{SoftMax}_i(u_i^\text{T} q_2)$ — эмбединг документа, Query q_2

Максимизация правдоподобия для классификации документов:

$\sum_d \sum_y \ln \text{SoftMax}_y(W_y v + b_y) \rightarrow \max$

Z.Yang, A.Smola et al. Hierarchical attention networks for document classification. 2016.

Внутреннее внимание или «самовнимание» (self-attention)

$\{x_t : t = 1, \ldots, n\}$ — входная последовательность токенов
$\{y_t : t = 1, \ldots, n\}$ — выходная последовательность

Идея:
модель обращает внимание на схожие токены из контекста;
не столь важно, генерируется новая последовательность или генерируются новые эмбединги исходной последовательности

Теперь h_i и h_t — эмбединги из одной последовательности:

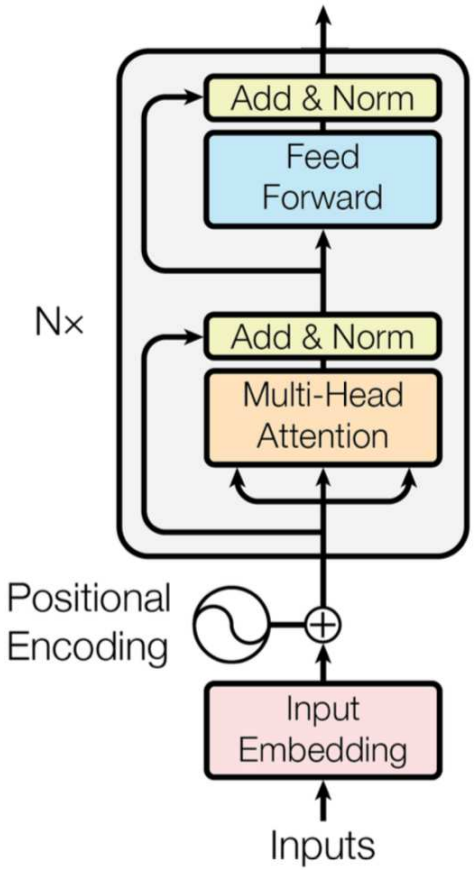
$$\left.\begin{aligned} a(h_i, h_t) &= (Kh_i)^{\mathsf{T}}(Qh_t) \\ \alpha_{ti} &= \mathrm{SoftMax}_i\, a(h_i, h_t) \\ c_t &= \textstyle\sum_i \alpha_{ti} V h_i \end{aligned}\right\} \quad c_t = \mathrm{SelfAttn}(h_t; Q, K, V)$$

$$y_t = f_y(h_t, y_{t-1}, c_t)$$

Vaswani et al. Attention is all you need. 2017.
Dichao Hu. An Introductory Survey on Attention Mechanisms in NLP Problems. 2018.

Transformer: архитектурные особенности кодировщика

1. Многомерное самовнимание, ℓ-й слой:
$$h_i^{k,\ell} = \text{SelfAttn}(h_i^\ell; Q^{k\ell}, K^{k\ell}, V^{k\ell})$$
2. Конкатенация и сохранение размерности:
$$h_i^{\ell+} = \left[h_i^{1,\ell} \cdots h_i^{K,\ell}\right] W$$
3. LayerNorm, 2 layers, и ещё раз LayerNorm:
$$h_i^{\ell+1} = \text{LN}(\text{FFN}(\text{LN}(h_i^{\ell+})))$$
$$\text{FFN}(x) = \text{ReLU}(W_1 x + b_1) W_2 + b_2$$
4. В FFN используются skip connections
5. Последовательно 6 блоков кодировщика
6. Кодируется информация о позициях



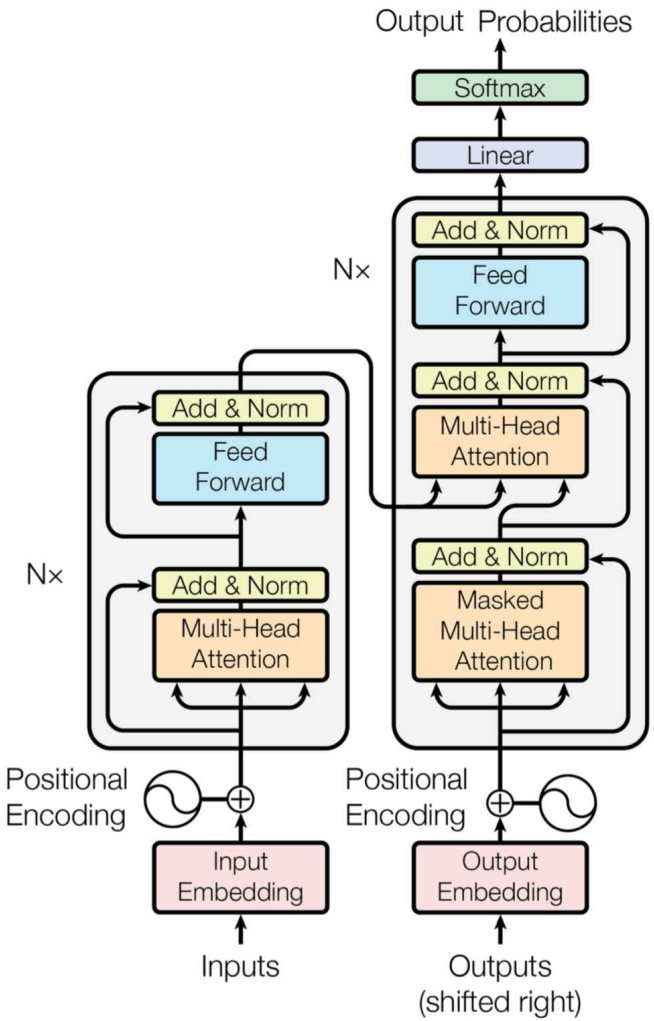
Vaswani et al. (Google) Attention is all you need. 2017.

Transformer: архитектурные особенности декодировщика

1. Дополнительный блок внимания, принимающий на входе выходные векторы кодировщика

2. Линейный предсказывающий слой:
$$y_t = \sigma(W_y h'_t + b_y)$$

3. Маскирование: предсказывающие слои не должны использовать последующие векторы $h'_{t+\delta}$



Vaswani et al. (Google) Attention is all you need. 2017.

Модель внимания Graph Attention Network (GAT)

Дано: граф $\langle V, E \rangle$

$h_i,\ i \in V$ — входные векторы признаков (или эмбединги) вершин

$h'_i,\ i \in V$ — выходные векторы вершин

$\mathcal{N}(t)$ — множество вершин $i \in V$ в окрестности вершины t

Функция сходства вершин i, t с параметрами u, v, W:

$a(h_i, h_t) = \exp\big(\text{LeakyReLU}(uWh_i + vWh_t)\big)$

$\alpha_{ti} = \underset{i \in \mathcal{N}(t)}{\text{norm}}\, a(h_i, h_t)$ — важность вершины i в контексте t

$c_t = \sum\limits_{i \in \mathcal{N}(t)} \alpha_{ti} W h_i$ — эмбединг контекста вершины t

$h'_t = \sigma(c_t)$ — выходной вектор для вершины t

Функция потерь определяется решаемой на графе задачей.

Petar Veličković et al. Graph Attention Networks. ICLR-2018.

Многомерное обобщение Multi-Head Attention для GAT

Дано: граф $\langle V, E \rangle$

$h_i,\ i \in V$ — входные векторы признаков (или эмбединги) вершин

$h'_i,\ i \in V$ — выходные векторы вершин

$\mathcal{N}(t)$ — множество вершин $i \in V$ в окрестности вершины t

K функций сходства вершин i, t с параметрами u^k, v^k, W^k:

$$a(h_i, h_t) = \exp\big(\text{LeakyReLU}(u^k W^k h_i + v^k W^k h_t)\big)$$

$\alpha_{ti} = \underset{i \in \mathcal{N}(t)}{\text{norm}}\, a(h_i, h_t)$ — важность вершины i в контексте t

$c_t^k = \sum\limits_{i \in \mathcal{N}(t)} \alpha_{ti} W^k h_i$ — эмбединг контекста вершины t

Два варианта выходного вектора для вершины t:

$h'_t = \text{concat}\big[\sigma(c_t^k)\big]_{k=1}^K$ — конкатенация

$h'_t = \sigma\big(\frac{1}{K}\sum_{k=1}^K c_t^k\big)$ — усреднение

Petar Veličković et al. Graph Attention Networks. ICLR-2018.

Многомерное обобщение Multi-Head Attention для GAT

Дано: граф $\langle V, E \rangle$

$h_i,\ i \in V$ — входные векторы признаков (или эмбединги) вершин

$h'_i,\ i \in V$ — выходные векторы вершин

Пример. $K = 3$ моделей внимания для преобразования $h_1 \to h'_1$

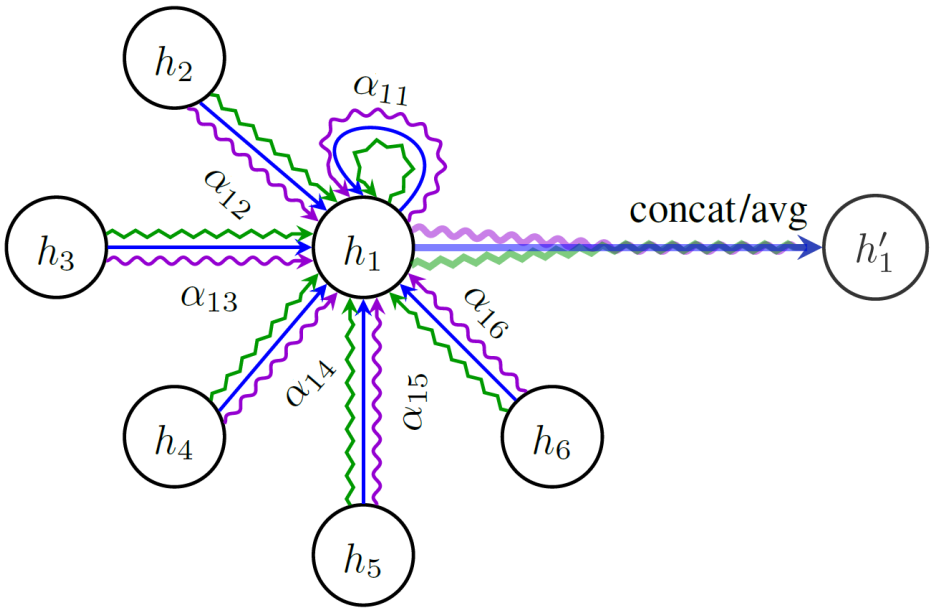
$$\alpha_{ti} = \underset{i \in \mathcal{N}(t)}{\text{norm}}\, a(h_i, h_t)$$

$$c_t^k = \sum_{i \in \mathcal{N}(t)} \alpha_{ti} W^k h_i$$

concat / average:

$$h'_t = \text{concat}\left[\sigma(c_t^k)\right]$$

$$h'_t = \sigma\left(\frac{1}{K}\sum_{k=1}^{K} c_t^k\right)$$



Petar Veličković et al. Graph Attention Networks. ICLR-2018.

Резюме

- Модели внимания сначала встраивались в RNN или CNN, но оказалось, что они самодостаточны
- Модель внимания работает точнее и быстрее RNN
- Легко обобщается на тексты, графы, изображения
- Доказано, что модель внимания multi-head self-attention (MHSA) эквивалентна свёрточной сети [Cordonnier, 2020]
- Модель внимания используются в наиболее продвинутых нейросетевых моделях BERT, GPT-2/3

Vaswani et al. Attention is all you need. 2017.
Dichao Hu. An Introductory Survey on Attention Mechanisms in NLP Problems. 2018.
Sneha Chaudhari et al. An Attentive Survey of Attention Models. 2019.
Cordonnier et al. On the relationship between self-attention and convolutional layers. 2020